Connect INA3221 on MCSBC-SVR to Meshtastic Device

The MCSBC-SVR has an onboard voltage/current sensor that reads battery and solar panel voltage and current, the INA3221. This sensor communicates over I2C and can be read by any computer, SBC or MCU that is capable of I2C and can load the appropriate library. When connecting the sensor to a Meshtastic device no coding is required (just two settings, described below).

Hardware Setup - I2C

The hardware setup is pretty simple. Just connect the 4 I2C wires between the MCSBC-SVR and your device: 3.3V, GND, SDA and SCL. These pins are labeled as such on the MCSBC-SVR. Your Meshtastic device might also have labels for these pins. Just connect these pins to the respective pins on each board (SDA to SDA, 3V3 to 3V3, etc). You can solder the wires directly to the boards or use the provided QWIIC connector on the MCSBC-SVR.

Plug-and-Play (no soldering)

QWIIC is an I2C connector supported by popular electronics companies like SparkFun (QWIIC) and Adafruit (STEMMA QT). QWIIC and STEMMA QT are compatible and use the same connector and pin order.

QWIIC on RAK Wireless Base Boards

If using a RAK Wireless base board you can attach their **RAK13009** QWIIC module to an open sensor slot. You could also use the **IO-XPNDR** module which not only has two QWIIC connectors but also exposes otherwise inaccessible pins. The IO-XPNDR attaches to the IO slot (compatible with the RAK19007, not the RAK19003). Either module provides a plug-and-play connection, no soldering required.

- RAK13009: https://store.rakwireless.com/products/rak13009-wisblock-qwiic-interface
- IO-XPNDR: https://www.etsy.com/listing/1747957894/io-xpndr-module-access-more-pins-on-your

Meshtastic Configuration for Viewing Voltage/Current in the App with Two Settings:

Set INA Address

In the Power module, set the device_battery_ina_address to "66".

- **CLI**: `meshtastic --set power.device_battery_ina_address 66`
- **Web Client**: go to Radio Config > Power and set INA219 Address to 66
- **Android app**: go to Radio Config > Power and set Device Battery INA Address to 66

Enable Power Measurements

In the Telemetry module, set power_measurement_enabled to true.

- **CLI**: `meshtastic --set telemetry.power_measurement_enabled true`
- **Web Client**: go to Module Config > Telemetry and set Power Measurement Enabled to Enabled/ON
- **Android app**: go to Module Config > Telemetry and set Power Measurement Enabled to Enabled/ON

You can change the Power Update Interval (`telemetry.power_update_interval`) if you want something other than the default time of 1800 seconds (30 minutes).

Enable MQTT From This Node

If you want to send voltage/current data over the mesh as Telemetry you need to allow it:

- **CLI**: `meshtastic --set lora.config_ok_to_mqtt true`
- **Web Client**: go to Radio Config > LoRa and set OK to MQTT to Enabled/ON
- **Android app**: go to Radio Config > LoRa and set OK to MQTT to Enabled/ON

This configuration only applies to Channels configured with the defaultpsk and eventpsk keys set in the Meshtastic Firmware; Channels with custom keys ignore this setting.

MQTT and Meshtastic

Meshtastic<->Network MQTT Bridge Node

To get Meshtastic messages and telemetry data onto your TCP/IP network you need another WiFi or ethernet-connected Meshtastic node that is configured to publish MQTT from the mesh to your network. This is often accomplished using an ESP32 node connected over WiFi.

Home Automation Server

You also need a device on your network with an MQTT broker and some way to view and manipulate the data. This is often accomplished using a home automation server. Node Red and Home Assistant are popular home automation server applications and they both offer integrations with Mosquitto, an MQTT broker. Node Red is "light" enough to run well on a Raspberry Pi 3B+ (even a Pi Zero 2, depending on what else you run). Home Assistant can run on a Pi 4 or Pi 5 (it could run on a Pi 3B+, again, depending). You could also use an old computer if you don't have a Raspberry Pi handy. If you want to save data to a database and use a Raspberry Pi, it's best practice to go with a Pi 4 or Pi 5 and save data to an attached SSD (not the SD Card, it will wear out too quickly).

Meshtastic on Linux with a LoRa Radio

The other option is to combine the above devices into one. Meshtastic supports "Linux Native" with **Mesthaticd** (a version for Linux). This is often accomplished using a Raspberry Pi with a LoRa "hat" (the radio hat must communicate over SPI, not UART). In theory, you could install your home automation server on the same device. Read more about Linux Native here:

- Linux Native Software: https://meshtastic.org/docs/software/linux-native/
- Linux Native Hardware: https://meshtastic.org/docs/hardware/devices/linux-native-hardware/

Refer to this Meshtastic doc for more details about MQTT and Meshtastic (scroll to the bottom for links to examples for Node Red and Home Assistant).

- **Meshtastic MQTT**: https://meshtastic.org/docs/software/integrations/mqtt/